

IN THE CLAIMS:

Please amend claims 1, 4, 6, 8-10, and 12-16.

Claims 2, 3, 5, 7, 11, 17 and 18, remain "as-is".

1. (Currently Amended). A system for component balancing in the processing of multiple transactional applications as implemented on a computer and wherein said processing normally follows a preset execution profile, said system comprising:

(a) means to establish and maintain response time goals for methods;

(b) means to delay other associated methods to optimize the processing of selected more-significant methods[[]] wherein said means to delay operates to change the execution profile of said multiple transactional applications.

2. (Original). The system of claim 1 which includes:

(c) means for sensing when an increased load is occurring in order to increase said delay applied to less-significant methods.

3. (Original). The system of claim 2 which includes:

(d) means for sensing when said load is decreasing in order to reduce the delay time applied to other associated methods.

4. (Currently Amended). A method for balancing and optimizing the processing of component transactional methods said method utilizing a computer for storing a software program and executing said method, comprising the steps of:

(a) selecting component transactional methods to gather runtime data from selected components;

(b) calculating statistical metrics between pairs of said transactional methods (A,B,C, . . . N);

(c) using ~~said~~ statistical significance tests on said metrics to select certain transactional methods for optimization and for delay in processing.

5. (Original). The method of claim 4 which includes the step of:

(d) establishing a goal of specified response time for each method (A,B,C, . . . N) selected from step (c).

6. (Currently Amended). The method of claim 5 wherein step (d) includes the steps of:

(d1) targeting specific groups of methods for delay[[]] wherein said delay will change the execution profile of said transactional methods;

(d2) setting specified response times as a goal for said specific groups of methods.

7. (Original). The method of claim 6 wherein step (d) includes the step of:

(d3) establishing a response time goal for each method from a setting of no delay in a method, to a maximum delay in a method.

8. (Currently Amended). The method of claim 7 which includes the step of:

(e)[[(i)]] graphically displaying individual response time for optimized methods against the response time goal set for a method.

9. (Currently Amended). In a component balancer system utilizing a computer for storing and executing a software program, a process for optimizing the sequence of processing component-based transactional applications, comprising the steps of:

- (a) selecting several methods ~~{for example, (A,B,C, . . . N)}~~ to be conditioned for analysis;
- (b) gathering runtime data from said selected methods in order to find statistical operating significance between selected pairs (AB, BA, AC, CA, BC, CB, . . .) of methods;
- (c) collecting data to get a representative workload involving said pairs (AB, BA, AC, CA, BC, CB, . . .) of said selected methods;
- (d) establishing an analysis report to determine when said method pairs (AB, BA, AC, CA, BC, CB, . . .) are processed to determine the average response time for processing when methods A,B,C, . . . N are run singly (non-overlapped) and when method pairs are run overlapped as AB, BA, AC, CA, BC, CB,

10. (Currently Amended). The method of claim 9 which includes the steps of:

(e) calculating a statistical number (F-value) which indicates the variance between average non-overlapped response times for A,B,C, . . . N and average response times for overlapped pairs of methods AB, BA, AC, CA, BC, CB,

(f) inquiring if the deviation in response times is below a threshold or if the average response time is below t milliseconds;

(g) selecting, above a threshold or an average response time, method calls having a deviation greater than t milliseconds;

(h) optimizing those method calls indicating a deviation greater than a threshold n involving an average response time greater than t milliseconds[[]] said optimization being effectuated by injecting code into a running application.

11. (Original). The method of claim 10 wherein step (h) includes the step of:

(h1) delaying the processing of one method in an overlapped pair of methods.

12. (Currently Amended). The ~~[[the]]~~ method of claim 11 ~~[[10]]~~ wherein step (h) further includes the step of:

(h2) removing a method if a period of time H, ~~such~~
~~as one hour,~~ elapses during which that method has
not been called.

13. (Currently Amended). In a component balancer system, utilizing a computer for storing and executing a software program, a method for optimizing the processing of component-based transactional applications, comprising the steps of:

- (a) securing a list of transactional applications to be optimized;
- (b) prioritizing said list according to a priority assigned to each transactional application;
- (c) accessing and capturing all or user selected components associated with said transactional applications;
- (d) analyzing which methods of which component should be optimized.

14. (Currently Amended). The method of claim 13 wherein step (d) includes the step of:

(d1) automatically optimizing the processing sequence of said transactional applications.

15. (Currently Amended). The method of claim 13 wherein step (d) includes the step of:

(d2) manually optimizing the processing sequence of said transactional applications.

16. (Currently Amended). A component balancer system for setting and managing response time goals for the processing of multiple component-based application methods (A,B,C, . . . N) said system utilizing a computer for storing and executing a software program, and wherein said application methods follow a fixed execution profile, said system comprising:

(a) means to discover and capture applications, transactional machines and components to be processed using a component runtime conditioner (CRC);

(b) means to analyze pairs of methods (AB, BA, AC, CA, BC, CB) to determine which method response times are affected by other methods;

(c) means to select those method pairs which show a substantial variance between the non-overlapped and the overlapped response times during the period involved with means (b) to analyze pairs;

(d) means to optimize the processing of selected method pairs[[]] by injecting code into a running application;

(e) means to apply delays in the processing of one associated method of a method pair[[]] said means operating to change the execution profile of said transactional applications.

17. (Original). The system of claim 16 where said means (e) to condition delays includes:

(e1) means to calculate said delay as a delay parameter using a fuzzy logic method to optimize said processing.

18. (Original). The system of claim 17 which includes:

(e2) means to adjust said delay increment according to the load on the system as sensed by the number of calls per second.